

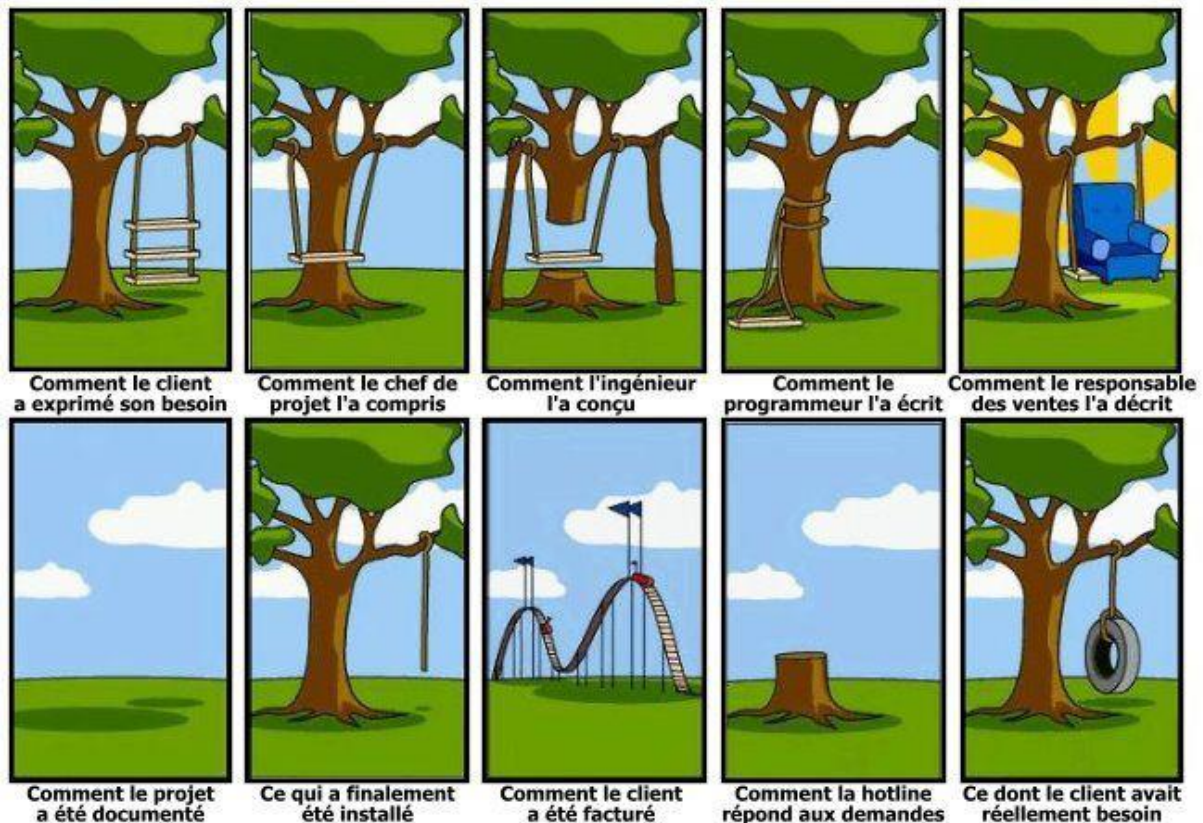
I. Phases de création d'un projet informatique

A. Cycle de vie d'un projet informatique

L'aboutissement d'un projet informatique est un processus complexe dont la réussite dépend de 3 paramètres : le temps, l'équipe de travail et le budget.

Ce processus contient les risques suivants :

- Non-respect du délai de livraison
- Dépassement du coût
- Non-respect des fonctionnalités demandées



Afin d'éviter ces risques, il est nécessaire de suivre les étapes suivantes :

■ **Préparation :**

- Etude générale du projet et de sa faisabilité
- Mise au point du plan d'organisation et de déroulement du projet
- Rédaction du cahier des charges

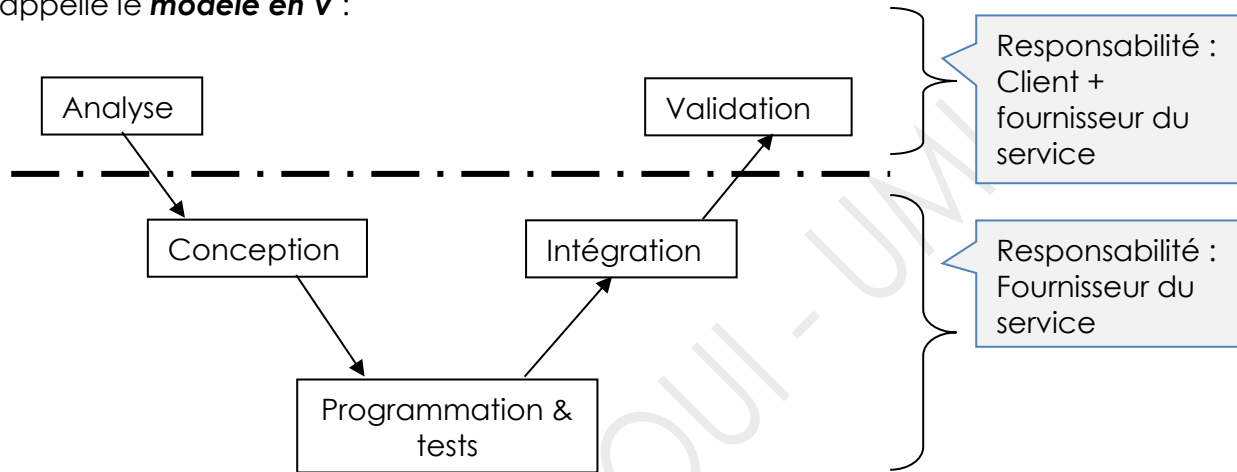
■ **Réalisation :**

- Analyse des besoins issus du cahier des charges : dégager la description technique et fonctionnelle de ce que doit faire le logiciel
- Conception des modèles ou modélisation : représentation graphique des besoins sous formes de schémas appelés « modèles »
- Codages ou programmation : traduire les modèles en algorithmes, puis en programmes écrits en langage de programmation (Java, Visual Basic, PHP, C++, ...)
- Tests : tester le bon fonctionnement de chaque composant du logiciel
- Intégration : rassembler tous les composants du logiciel pour former le produit final

- **Validation** : Vérifier si le produit est conforme aux besoins exprimés dans le cahier des charges
- **Documentation & formation** : Rédiger les documents nécessaires à l'utilisation du logiciel, et former les prochains utilisateurs sur ce dernier si c'est nécessaire
- **Livraison** : Installation du livrable chez le client

La phase de réalisation est l'étape concrète de création du logiciel.

Le cycle de vie de ce dernier est souvent représenté dans cette phase par ce qu'on appelle le **modèle en V** :



B. Pourquoi modéliser

La conception, ou modélisation, consiste à schématiser, ou créer une représentation virtuelle, des besoins exprimés dans le cahier des charges.

Cela permet de :

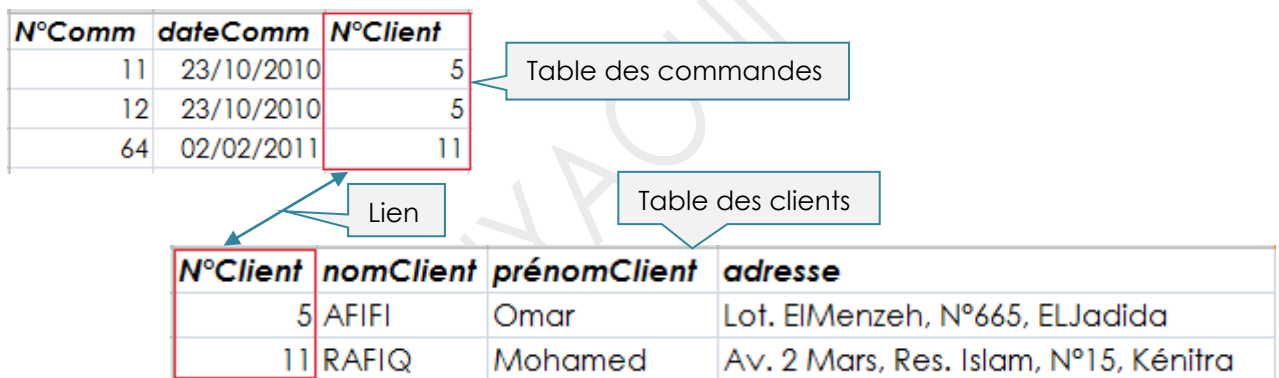
- Minimiser au maximum les erreurs lors de la phase de programmation, car durant cette phase, les erreurs sont beaucoup plus coûteuses en temps, donc en argent.
- Assurer la cohérence et éviter la redondance des données
- Assurer la conformité du logiciel aux exigences du client

Exemple : Gestion de commandes

N°Comm	dateComm	N°Client	nomClient	prénomClient	adresse
11	23/10/2010	5	AFIFI	Omar	Lot. ElMenzeh, N°665, ELJadida
12	23/10/2010	5	AFIFI	Omar	Lot. ElMenzeh, N°665, ELJadida
64	02/02/2011	11	RAFIQ	Mohamed	Av. 2 Mars, Res. Islam, N°15, Kénitra

Pour **2** commandes, les informations du client **N°5** sont saisies **2** fois.

Il serait plus logique d'avoir deux tableaux avec un lien entre les deux :



Les questions qui se posent donc sont :

- Quel champ doit être placé dans quelle table ?
- Quel est le champ qui jouera le rôle du lien entre les tables ?

Pour répondre à ce genre de questions, il est nécessaire de recourir à une solution d'analyse et de conception qui soit **méthodique et standardisée**. Les plus utilisées sont **MERISE** et **UML**.

C. Introduction à UML

1. Définition

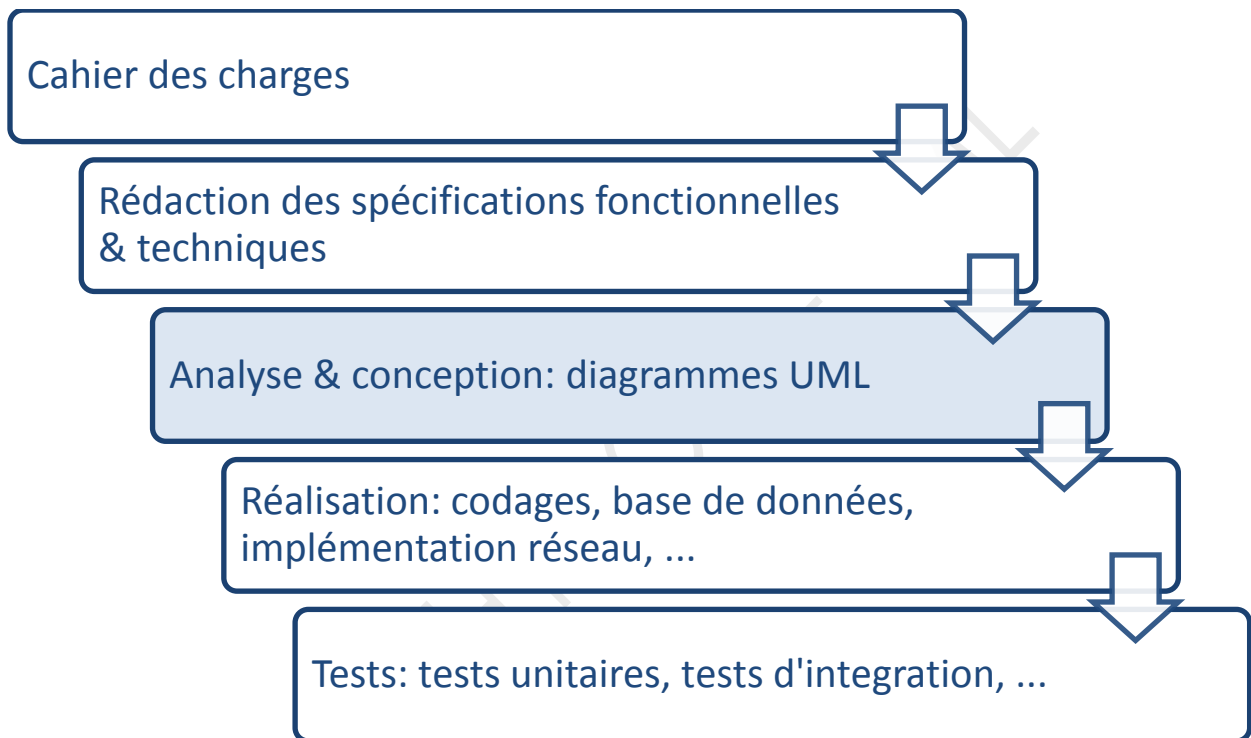
UML (*Unified Modeling Language*) est un langage, ou notation, de modélisation graphique à base de diagrammes, utilisé dans la conception & développement orientés objet.

UML est un langage standardisé par l'*Object Management Group*

(<http://www.uml.org>)

La dernière version diffusée par l'OMG est UML 2.4.1 depuis août 2011.

2. Quand utiliser UML



3. Aperçus sur les diagrammes d'UML

Les diagrammes UML se répartissent en **2** grands groupes :

- Diagrammes **structurels ou statiques**
- Diagrammes **comportementaux**

Diagrammes structurels ou statiques

- Diagramme de classes (Class diagram)
- Diagramme d'objets (Object diagram)
- Diagramme de composants (Component diagram)
- Diagramme de déploiement (Deployment diagram)
- Diagramme des paquetages (Package diagram)
- Diagramme de structure composite (Composite structure diagram)
- Diagramme de profils (Profil Diagram)

Diagrammes comportementaux

- Diagramme des cas d'utilisation (use-cases ou Use Case Diagram)
- Diagramme d'états-transitions (State Machine Diagram)
- Diagramme d'activité (Activity Diagram)
- Diagramme de séquence (Sequence Diagram)
- Diagramme de communication (Communication Diagram)
- Diagramme global d'interaction (Interaction Overview Diagram)
- Diagramme de temps (Timing Diagram)

(Les 4 derniers sont souvent appelés *Diagrammes d'interaction ou dynamiques*)

Grâce aux outils de modélisation UML, il est également possible de générer automatiquement des parties de code (par exemple en langage Java) ou des supports de documentation à partir des modèles réalisés.

II. Diagrammes d'analyse

A. Modélisation des besoins : Diagramme des cas d'utilisation

1. Définition

Définition : Besoins fonctionnels / techniques

Tirés du cahier des charges, les **besoins** ou **exigences** sont l'expression documentée de ce qu'une application informatique **devrait faire** ou **doit être**

Deux types d'exigences :

→ **Exigences fonctionnelles**

Décrivent le système sur le plan **fonctionnel**, c.à.d. ce que le système doit faire.

Elles reflètent les besoins exprimés par le client dans le cahier de charges, et elles ne sont pas négociables.

En général, ce genre d'exigences répond à la question : « Pourquoi l'utilisateur aurait-il besoin de l'application ? »

→ **Exigences non fonctionnelles (ou techniques)**

Décrivent le système sur le plan **technique**, c.à.d. la manière dont il exécute les fonctionnalités que l'utilisateur lui demande

Elles reflètent généralement les contraintes techniques imposées par l'architecture logicielle du futur produit : le choix de la plateforme et du langage de programmation, le type de système d'exploitation visé (*Unix, Windows, Android, ...*), le déploiement sur le réseau (*machine locale, intranet, internet, ...*), etc.

Ce genre d'exigences peut être exprimé par le client ou par le développeur

Exemple : Besoins fonctionnels / techniques

Etude de cas : Sites internet pour réseaux sociaux

- Pouvoir ouvrir plusieurs conversations instantanées à la fois : figure parmi les services les plus élémentaires et indispensables d'une application de réseau social, ce service doit donc être considéré comme une **exigence fonctionnelle**

- Pouvoir ouvrir plus de 50 conversations instantanées à la fois : ceci peut être réalisable ou non, selon l'architecture technique de l'application (plateforme, environnement de développement, langage de programmation, ...), ce service doit donc être considéré comme une **exigence non fonctionnelle**

2. Analyse des besoins

Exemple : Besoins fonctionnels / techniques

Etude de cas : Guichet automatique bancaire (GAB)

A partir du texte du cahier de charges de l'application du GAB, on peut déduire les exigences suivantes :

▪ Exigences fonctionnelles

- L'utilisateur doit pouvoir consulter le solde de son compte
- L'utilisateur doit pouvoir extraire un mini-relevé de son compte
- L'utilisateur doit pouvoir retirer de l'argent de son compte
- ...

▪ Exigences non fonctionnelles

- L'opération d'extraction d'un mini-relevé de compte bancaire ne doit pas durer plus de 5sec
- Lors de l'opération de retrait d'argent, si l'utilisateur tape un montant supérieur au montant maximal, un message d'erreur doit lui indiquer le montant à ne pas dépasser
- ...

Etude de cas : Gestion commerciale

[.....

Pour passer une commande le client s'adresse au réceptionniste, celui-ci passe la commande (commande normale ou express) si le client existe déjà, sinon il doit d'abord le créer.

Le comptable ou le directeur des ventes peuvent enregistrer la facture de chaque commande dont le paiement est dûment réglé.

Le livreur doit enregistrer chaque livraison qu'il effectue.

Pour les commandes dont la liste de produits n'existe pas en quantité suffisante, le directeur des ventes peut aussi gérer le stock pour éviter toute rupture de produits.

La gestion du stock contient les fonctionnalités suivantes :

Gérer un produit, et enregistrer une demande d'alimentation du stock

La gestion d'un produit contient les fonctionnalités suivantes :

Ajouter, supprimer et mettre à jour un produit

Chaque utilisateur du système, doit d'abord s'authentifier avant de pouvoir accéder à n'importe quelle fonctionnalité.

.....]

Travail à faire :

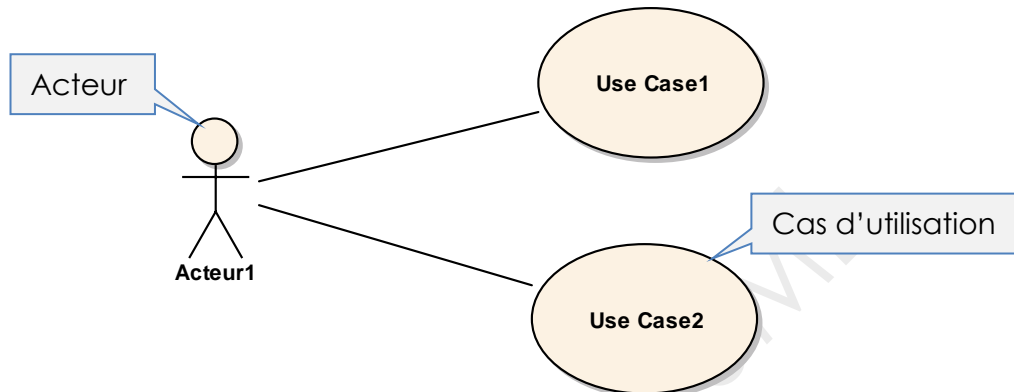
- Rédiger la liste des exigences fonctionnelles

3. Modélisation des besoins : Diagramme des Cas d'utilisation

Définition : Diagramme des cas d'utilisation

Le diagramme des cas d'utilisation – *use case diagram* – est considéré comme le point d'entrée de la phase d'analyse, il permet de modéliser toutes les fonctionnalités que doit fournir le système.

Le formalise graphique du DCU est principalement basé sur les **acteurs** et les **cas d'utilisation**

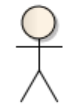


Chaque exigence ou besoin **fonctionnel** identifié dans la phase de rédaction des exigences, est représenté par un use-case.

4. Réaliser le UCD

→ Les acteurs

Définition : Les acteurs



Les **acteurs** sont soit (le plus souvent) des utilisateurs humains du système, soit d'autres systèmes informatiques ou hardwares qui vont communiquer avec le système.

Exemples :

Gestionnaire de stock, chargé de clientèle, administrateur, ...

Exemple

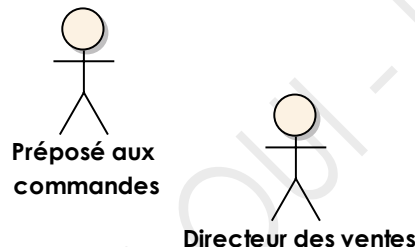
Fragment d'un cahier de charges : Gestion commerciale

[...

Le directeur des ventes est un préposé aux commandes, mais avec un pouvoir supplémentaire : en plus de pouvoir passer et suivre une commande, il peut gérer le stock.

Par contre, un simple préposé aux commandes n'a pas le droit d'accéder à la gestion du stock

...]



→ Les relations entre acteurs

La seule relation possible entre deux acteurs est le lien d'héritage (ou généralisation) :

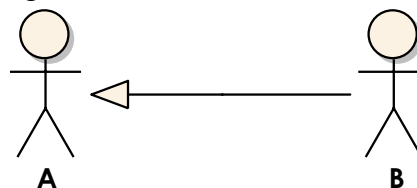
Définition : L'héritage entre acteurs

On dit qu'un acteur **A** est une **généralisation** d'un acteur **B**, on dit aussi : l'acteur **B** est un cas particulier de l'acteur **A**, quand :

Tous les cas d'utilisation appartenant à l'acteur **A**, appartiennent aussi à **B**

On dit alors que **B hérite de A**

Notation graphique de l'héritage :



Exemple

Etude de cas : Gestion commerciale

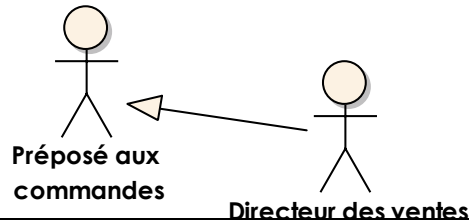
[...

Le préposé aux commandes peut passer une commande, et suivre une commande. Le directeur des ventes **est lui aussi** un préposé aux commandes, mais avec un **pouvoir supplémentaire** : en plus de pouvoir passer et suivre une commande, il peut gérer le stock.

Par contre, un simple préposé aux commandes n'a pas le droit d'accéder à la gestion du stock

...]

Acteurs :



→ Les cas d'utilisation

Définition : Les cas d'utilisation

Chaque **cas d'utilisation** représente **une exigence fonctionnelle**.
 Exemples :
Retirer argent, Passer une commande, Supprimer un produit, ...

Exemple**Cas d'utilisation : Retirer argent****Etude de cas : GAB**

Retirer argent

Un cas d'utilisation doit **obligatoirement** avoir les éléments suivants :

- Un nom

Le nom d'un cas d'utilisation commence généralement par un verbe à l'infinitif

**IMPORTANT**

- Deux ou plusieurs cas d'utilisation ne peuvent avoir le même nom.
- Un cas d'utilisation ne peut avoir plusieurs noms.

- Un ou plusieurs acteurs déclencheurs

Un use-case doit avoir au moins un seul acteur déclencheur.

Définition : Acteur déclencheurUn acteur déclencheur d'un use-case **A** est un acteur qui utilise le système pour accéder à la fonctionnalité représentée par le cas **A**.**Exemple****Etude de cas : Gestion de centre de formation**

[...

Le directeur pédagogique peut ajouter ou supprimer des modules tout au long de l'année scolaire

...]

On peut donc dire que l'acteur **directeur pédagogique** est l'acteur déclencheur des cas d'utilisation **ajouter un module** et **supprimer un module****REMARQUES**

- Un acteur peut posséder plusieurs cas d'utilisation.
- Un use-case peut avoir plusieurs acteurs déclencheurs.

- Des scénarios

Un cas d'utilisation doit obligatoirement avoir un scénario principal (**unique**), et peut avoir un ou plusieurs scénarios secondaires.

Définition : Scénario / Scénario principal / Scénario secondaire

- **Scénario** : décrit (sous forme de texte) les échanges d'évènements entre l'acteur et le système.
- **Scénario principal** : représente le déroulement de la réalisation avec succès du use-case. Il doit se terminer par la fin attendue de la part de l'acteur déclencheur.
- **Scénario principal** : permet de préciser la conduite à tenir par le système en cas d'anomalies ou d'erreurs.
- **Scénario alternatif** : scénario sans erreur mais qui décrit une fin alternative à celle du scénario principal.

Exemple

Etude de cas : GAB

Cas d'utilisation : « Retirer argent »

Scénario principal : Argent retiré

Scénario d'erreur N°1 : Carte invalide,

fin du scénario : carte éjectée

Scénario d'erreur N°2 : Mot de passe invalide, fin du scénario : carte aspirée

Scénario d'erreur N°3 : Caisse épuisée,

fin du scénario : carte éjectée

Scénario d'erreur N°4 : Solde insuffisant,

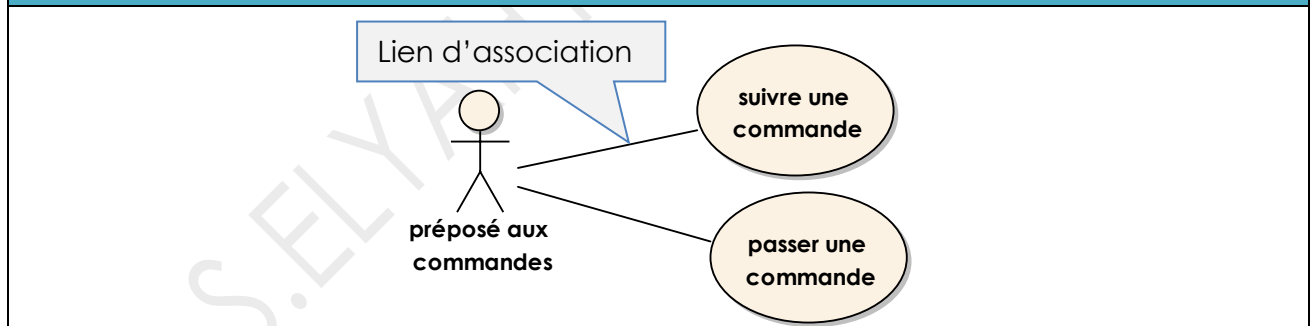
fin du scénario : carte éjectée

→ La relation d'association entre acteur et cas d'utilisation

Chaque acteur **doit être lié** par un lien d'association avec tous les use-cases qu'il déclenche, ce lien est représenté par un trait **continu**

Exemple

Etude de cas : Gestion commerciale



Etude de cas : Guichet automatique bancaire

[.....

Le porteur de carte (soit client de la banque soit client d'une autre banque) peut retirer de l'argent.

Si le porteur est un client de la banque, il peut aussi consulter son compte, ou retirer un mini-relevé bancaire.

Il peut aussi (client de la banque) modifier ses paramètres de sécurité
.....]

Travail à faire :

- Identifier les acteurs et leurs cas d'utilisation associés
- Réaliser le diagramme des cas d'utilisation

→ **Les relations entre cas d'utilisation**

UML définit **3** principaux types de relations entre cas d'utilisation :

- L'héritage

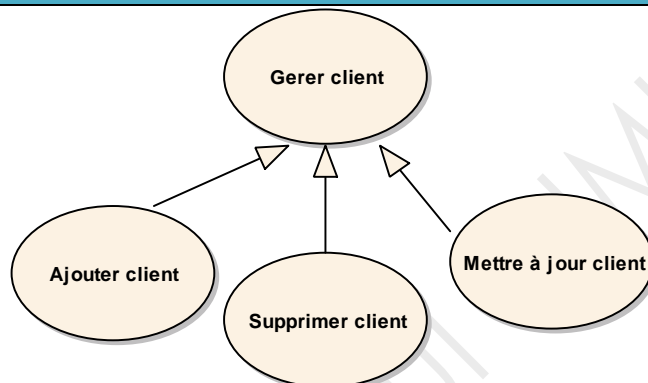
Définition

On dit qu'un use-case **B** hérite d'un autre use-case **A**, quand **A** représente une généralisation de **B**

Graphiquement, l'héritage entre cas d'utilisation est noté de la même manière que celui des acteurs

Exemple

Etude de cas : Gestion commerciale

**IMPORTANT**

- L'association d'héritage entre use-cases est utilisée principalement dans un souci d'organisation graphique du UCD
- En général, un use-case générique (dans l'exemple : « *Gérer client* ») n'a pas d'existence propre à lui, d'ailleurs il ne peut pas avoir de scénarios, il ne se réalise qu'à travers l'un de ses use-case dérivés

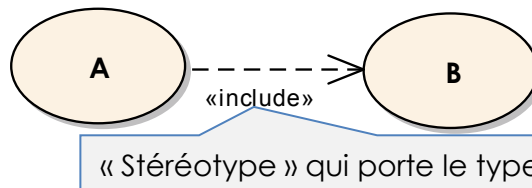
- L'inclusion

Définition : « include »

On dit qu'un use-case **B** est inclus d'un autre use-case **A**, quand le déroulement de **B** représente **toujours** une partie du déroulement de **A**

Autrement dit, si **B** est inclus dans **A**, alors l'acteur **ne peut réaliser le use-case A, que si B est terminé correctement**.

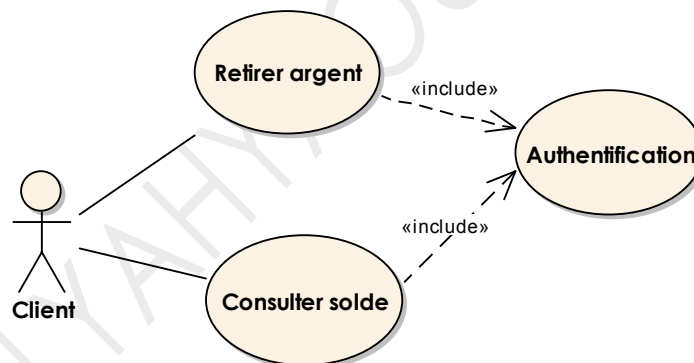
Graphiquement, la relation d'inclusion est notée par une flèche discontinue allant de **A** à **B**, accompagnée du stéréotype « **include** »


Exemple
Etude de cas : GAB

[...

Pour pouvoir retirer de l'argent ou simplement consulter son solde, un client doit **toujours** s'authentifier en insérant sa carte guichet, le système analyse la carte et demande le mot de passe ...

...]

**REMARQUES**

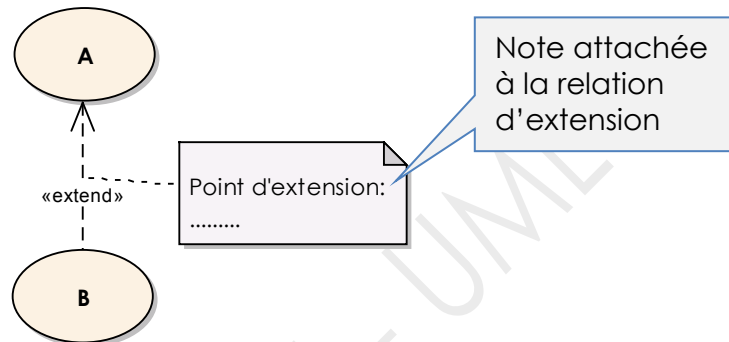
- Si un use-case **B** est inclus dans un autre **A**, alors en général **B** n'est pas à proprement dit **un vrai cas d'utilisation**, car pour l'utilisateur du logiciel le cas d'utilisation **B** ne représente pas une fonctionnalité du système, donc le fait d'associer l'acteur au use-case **B** serait **une erreur**.
- Une autre erreur courante est d'utiliser la relation « *include* » pour faire « le découpage » d'un cas d'utilisation en plusieurs « sous cas d'utilisation » qui s'enchaînent en fonction de certains critères.

- L'extension

Définition : « extends »

On dit qu'un use-case **B** étend un autre use-case **A**, quand **B** peut être appelé au cours de l'exécution de **A** si une condition est vérifiée (**donc pas toujours**). Cette condition est appelée **point d'extension**.

Graphiquement, la relation d'inclusion est notée par une flèche discontinue allant de **B** à **A**, accompagnée du stéréotype « extends », cette flèche **doit obligatoirement** être commentée par une note indiquant le point d'extension



i REMARQUE

On peut attacher une note à n'importe quel élément graphique dans un diagramme UML

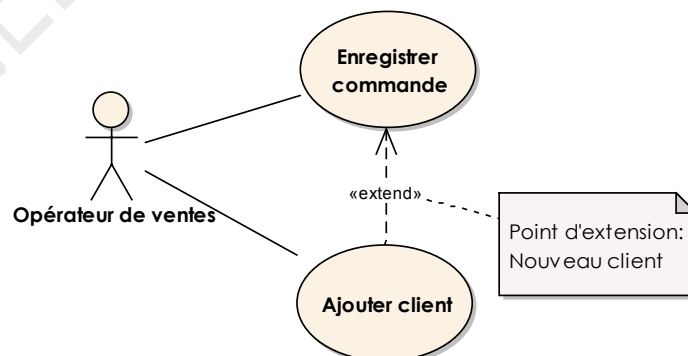
Exemple 1**Etude de cas : Gestion commerciale**

[...

L'opérateur de ventes peut enregistrer les commandes des clients, et chaque commande doit appartenir à un client parmi la liste des clients enregistrés dans le système.

S'il s'agit d'un nouveau client, l'opérateur doit pouvoir le créer avant de lui enregistrer sa commande, ...

...]

**Exemple 2****Etude de cas : Gestion électronique des actes de ventes immobilières**

[...

Le notaire doit pouvoir disposer d'une nouvelle fenêtre pour la création de chaque contrat.

Chaque contrat contient une liste de vendeurs et une liste d'acquéreurs, le notaire doit donc pouvoir choisir un client parmi la liste des clients (si le client existe déjà), et l'ajouter

soit dans la liste des vendeurs, soit dans celle des acquéreurs.

S'il s'agit d'un nouveau client, il doit d'abord le créer.

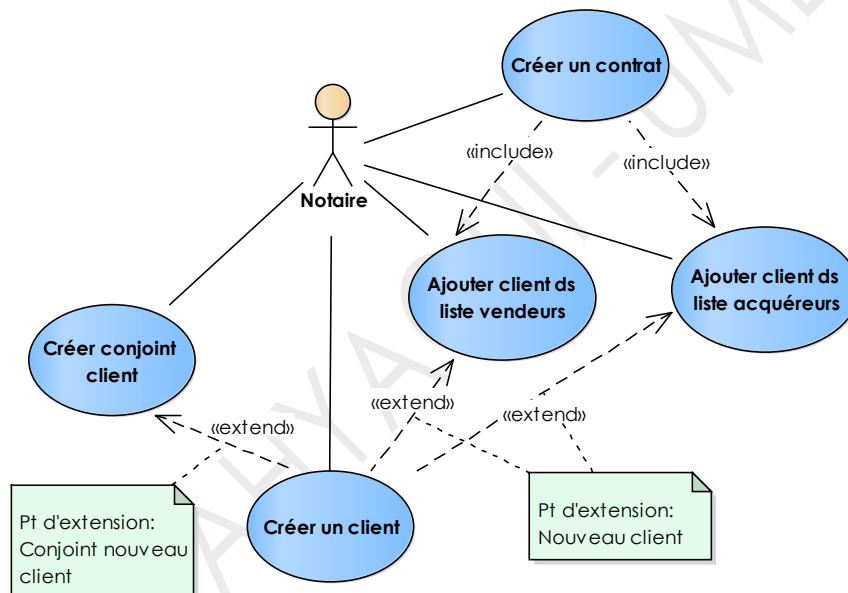
Lors de la création d'un client, (si le client est marié) le notaire doit pouvoir spécifier les informations relatives au conjoint du client. Si le conjoint du client n'existe pas, il doit d'abord le créer.

...]

Exigences fonctionnelles :

- Créer un contrat
- Ajouter un client dans liste des vendeurs
- Ajouter un client dans liste des acquéreurs
- Créer un client
- Créer le conjoint d'un client

Diagramme des cas d'utilisation :



Etudes de cas :

Guichet automatique bancaire

[.....

N'importe quel porteur de carte bancaire peut retirer de l'argent, ou (s'il le souhaite) retirer de l'argent avec un reçu.

Si le porteur est un client de la banque, il doit pouvoir accéder aussi aux services suivants (en plus du retrait d'argent) :

Déposer de l'argent, retirer un mini-relevé bancaire de son compte, consulter son solde, payer la facture d'abonnement du téléphone fixe et modifier ses paramètres de sécurité.

Un porteur de carte ne peut accéder à aucun service du GAB sans être d'abord authentifié.

.....]

Travail à faire :

- **Identifier les acteurs et leurs cas d'utilisation associés**
- **Réaliser le diagramme des cas d'utilisation**

Gestion commerciale

[.....

Pour passer une commande le client s'adresse au réceptionniste, celui-ci passe la commande (commande normale ou express) si le client existe déjà, sinon il doit d'abord le créer.

Le comptable et le directeur des ventes peuvent enregistrer la facture de chaque commande dont le paiement est dûment réglé.

Le livreur doit enregistrer chaque livraison qu'il effectue.

Pour les commandes dont la liste de produits n'existe pas en quantité suffisante, le directeur des ventes peut aussi gérer le stock pour éviter toute rupture de produits.

La gestion du stock contient les fonctionnalités suivantes :

Gérer un produit, et enregistrer une demande d'alimentation du stock

La gestion d'un produit contient les fonctionnalités suivantes :

Ajouter, supprimer et mettre à jour un produit

Chaque utilisateur du système, doit d'abord s'authentifier avant de pouvoir accéder à n'importe quelle fonctionnalité.

.....]

Travail à faire :

- **Identifier les acteurs et leurs cas d'utilisation associés**
- **Réaliser le diagramme des cas d'utilisation**

Gestion de réservations d'hôtel

[.....

La direction d'un hôtel souhaite créer un système informatique pour la gestion des chambres et des réservations.

Ce système sera composé de deux parties, un site web pour les réservations des clients par internet, et une application Windows pour les employés de l'hôtel.

Un visiteur normal du site peut uniquement afficher la liste des chambres.

S'il s'agit d'un client, il peut modifier ses informations (login, password, nom, numéro de téléphone, ...), afficher la liste des chambres, passer une réservation ou bien annuler sa réservation.

Au cas où c'est le réceptionniste qui enregistre la réservation pour le client, il (réceptionniste) doit d'abord ajouter le client si ce dernier n'est pas encore enregistré dans le système.

De même, si c'est un nouveau client qui passe la réservation, il doit d'abord créer un compte.

Le réceptionniste peut également afficher la liste des chambres disponibles, la liste des réservations, et la liste des clients.

Le gérant quant à lui, peut en plus de cela, gérer la liste des clients.

La gestion de la liste des clients comprend l'ajout, la mise à jour, et la suppression de clients.

Chaque utilisateur du système doit s'authentifier, sauf les visiteurs normaux, pour l'affichage de la liste des chambres.

.....]

Travail à faire :

- Identifier les acteurs et leurs cas d'utilisation associés
- Réaliser le diagramme des cas d'utilisation

Gestion de médiathèque

[.....

Le système d'informations d'une médiathèque de livres informatiques comporte deux interfaces graphiques : un site internet pour les clients, et une application Windows pour les bibliothécaires et l'administrateur.

Le site internet est ouvert à tout internaute pour parcourir la liste des livres, afficher le sommaire d'un livre ou encore, créer un compte d'adhérent.

Si l'internaute est un adhérent, il peut aussi (après s'être authentifié) créer un panier, enregistrer un panier pour une modification ultérieure, ouvrir un panier déjà créé, valider l'achat d'un panier (un panier contient au moins un livre).

Il peut aussi (l'adhérent) modifier les informations de son compte (nom, prénom, ...) ou signaler au système qu'il a oublié son mot de passe.

Le bibliothécaire (utilisateur de l'interface Windows) doit pouvoir ajouter ou supprimer un livre, ajouter ou supprimer une catégorie de livres (développement, base de données, systèmes & réseaux, ...), ou encore afficher la liste des paniers validés, et la liste des paniers livrés.

Un livre ne doit pas exister dans la médiathèque, sans appartenir à une catégorie.

Donc lors de l'ajout d'un livre, sa catégorie doit être spécifiée. S'il s'agit d'une nouvelle catégorie, elle doit être d'abord créée.

L'administrateur (lui aussi utilisateur de l'interface Windows) doit pouvoir afficher tous les comptes utilisateurs (qu'ils soient de type adhérent ou bibliothécaire), supprimer ou créer un compte bibliothécaire, ou modifier les informations de son propre compte à lui (nom, prénom, mot de passe, ...).

Il doit aussi (l'administrateur) pouvoir afficher la liste des paniers validés, et la liste des paniers livrés, et afficher la recette du jour et la recette du mois.

*Bibliothécaires et administrateur doivent aussi s'authentifier.
.....]*

Travail à faire :

- **Identifier les acteurs et leurs cas d'utilisation associés**
- **Réaliser le diagramme des cas d'utilisation**

S.ELYAHYAOU - UML